Examen 1. Preguntas y respuestas comentadas

1. Estado es una clase abstracta que sirve para modificar el estado de una cama, ya sea ocupado o libre.

e) Completamente en desacuerdo.

Estado (debía ser Estado_Cama) es una clase abstracta pero NO modifica el estado de ninguna cama. Su papel es servir de interfaz para las clases que se ocupan individualmente del comportamiento de cada estado posible del objeto cama. La clase Estado (de algo) permite establecer un contrato común con las clases que se ocupan del comportamiento de cada estado. Es decir, permite homogenizar la diversidad de los diferentes estados. Ver Patrón Estado. (El enunciado de la pregunta fue tomado de un estudiante).

2. El ratón mueve las figuras que están en la ventana. Por esta causa, cuando se pulsa, el ratón debe conocer la posición de cada figura para saber si puede actuar sobre ella.

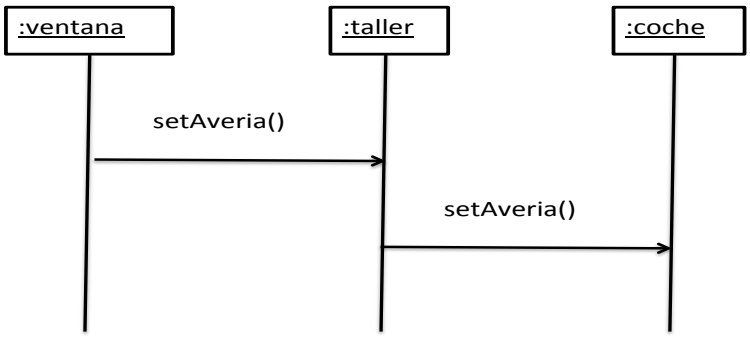
e) Completamente en desacuerdo.

Porque el ratón debe conocer a cada figura y el interior de cada una de ellas. Por tanto, el ratón tiene una elevada dependencia (unívoca) con las figuras. Para reducir esta dependencia y convertirla en una dependencia indiferente (ambigua) el ratón debe emitir su posición y que cada figura reacciones según le afecte o no.

4. El patrón cadena de responsabilidades sirve para:

a) Reducir el grado de sorpresa del objeto que le toca reaccionar en una lista.
b) Esconder una lista.
c) Definir la responsabilidad de cada objeto de una lista.
d) Conocer con más detalle el manejo de una lista.

a) Falso. Reducir el grado de sorpresa de un elemento software con respecto a otro es una cualidad adecuada del diseño para reducir su complejidad. Pero el patrón cadena de responsabilidades NO reduce el grado de sorpresa del objeto que le toca reaccionar en una lista. El propósito de este patrón es reducir el grado de sorpresa del elemento que utiliza la lista.
b) Verdadero. El patrón reduce el grado de sorpresa del elemento que utiliza la lista escondiendo la lista a sus ojos. Este elemento ignora que hay una lista, solo conoce al primer miembro de la lista; el resto de la lista queda oculto detrás.
c) Falso. El patrón sirve para reducir el grado de sorpresa del elemento que utiliza la lista; establece una relación ambigua con la lista.
d) Falso. El patrón sirve para reducir el grado de sorpresa del elemento que utiliza la lista; establece una relación ambigua con la lista.



3. El diseño de la figura anterior:

a) **No cumple el Principio de Ocultación de Información porque Ventana conoce el interior de Taller.**
b) **No cumple el Principio de Ocultación de Información porque setAvería() es un método público.**
c) **No cumple el Principio de Ocultación de Información porque Taller conoce el interior de Coche.**
d) **Cumple el Principio de Ocultación de Información porque el atributo Avería es privado en Taller y Coche.**

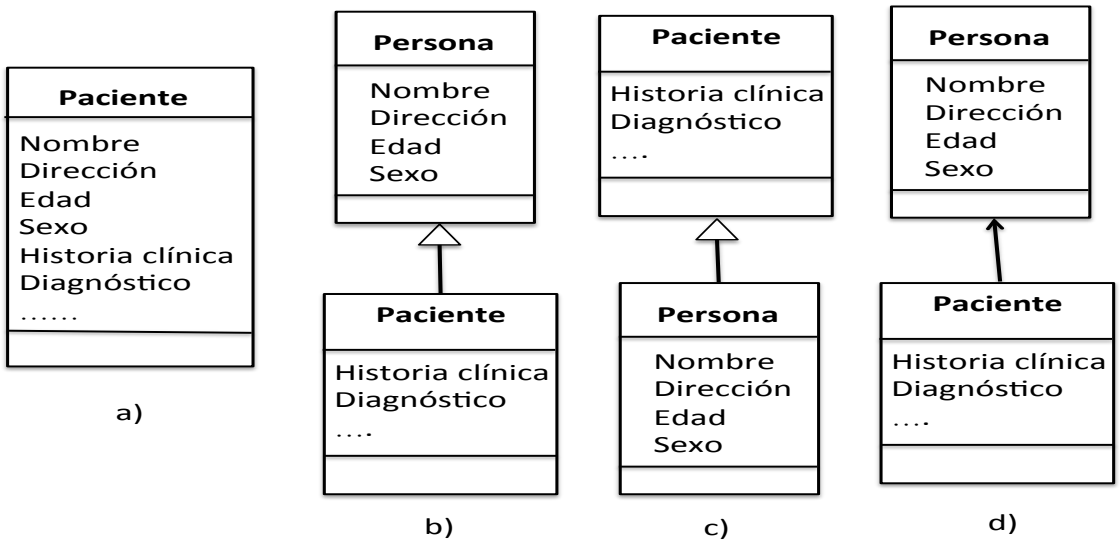
a) *Podría ser la respuesta acertada porque efectivamente, ventana conoce el interior de Taller, pero hay que considerar que Ventana pudiera ser el formulario asociado con Taller para tomar información del exterior. En este caso Ventana tiene que conocer el interior de Taller y no viola el Principio de Ocultación porque Ventana (aunque es una clase separada) forma parte de Taller. Sigamos analizando el resto de alternativas.*

b) *Falso. La condición de privado o público de un método o un atributo no influye en el cumplimiento del Principio de Ocultación.*

c) *Verdadero. Se incumple el Principio de Ocultación porque Taller conoce el interior de Coche. La información de la avería debe llegar a Coche desde el exterior mediante una ventana o formulario asociado a Coche.*

d) *Falso. La condición de privado o público de un método o un atributo no influye en el cumplimiento del Principio de Ocultación.*

Evaluando el conjunto de alternativas, la máxima certeza es la (c).



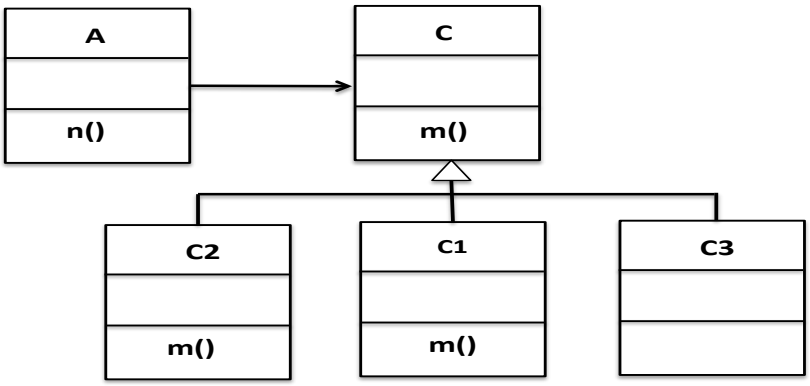
5. ¿Qué diseño de la figura anterior considera mejor?

a) *Este diseño parece el más simple porque tiene menos clases, pero es una falsa imagen. La complejidad de persona y paciente está presente, solo que apelotonado en una clase. Esto impide distribuir trabajo (alguien que se ocupe de persona y alguien que se ocupe de paciente) y además dificulta la modificación y evolución del sistema. Los conceptos de persona y paciente son diferentes, por tanto deben estar separados.*

b) *La relación de Paciente "hereda de" Persona parece adecuada en virtud de la interpretación de la herencia software como una relación "es un". Sin embargo, en el presente caso sólo es una relación circunstancial: el paciente es una persona porque se trata de un hospital, pero el concepto de paciente no implica ser persona. Por tanto, es poco conveniente ligarlos tan fuertemente en el software. Resulta mejor una relación se asociación mucho más débil que la herencia. Respuesta d).*

c) *La relación Persona es un Paciente expresa que toda persona es un paciente, por tanto está equivocado.*

d) *La solución software adecuada es asociar Paciente con Persona porque expresa mejor la diferencia entre estos conceptos y porque es una relación más débil, más separable que la relación de herencia.*



6. El diseño de la figura anterior es:

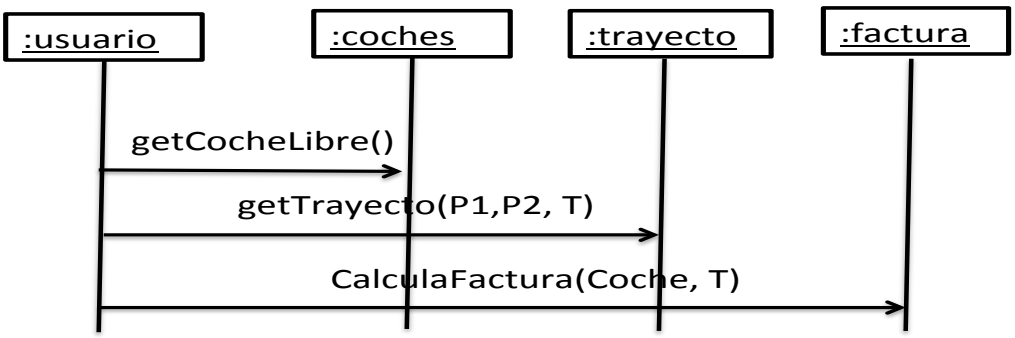
a) **Adecuado porque reduce la sorpresa de n() respecto a m().**
b) **Inadecuado porque utiliza demasiadas clases.**
c) **Adecuado porque C3 utiliza el mismo m() que C().**
d) **Inadecuado porque m() de C tiene que tener código para que C3 sea concreta.**

a) *Cierto, porque n() tiene una relación de dependencia indiferente (ambigua) con los métodos m().*

b) *Falso. La cantidad de clases por sí misma no es un índice para evaluar un diseño.*

c) *Falso. La clase C3 sobra porque es idéntica a la clase C.*

d) *Falso, porque el argumento utilizado (m() de C tiene que tener código para que C3 sea concreta) no justifica que el diseño sea inadecuado.*



7. El diseño de la figura anterior es:

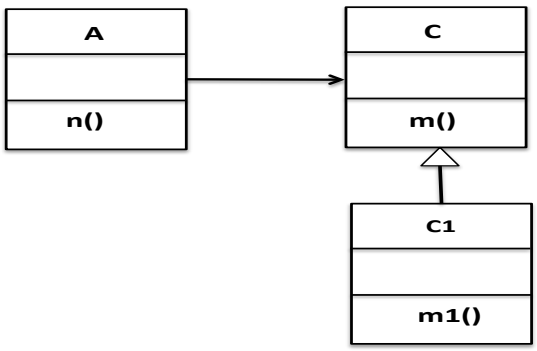
a) Adecuado porque es simple.
b) Inadecuado porque :usuario está sobrecargado.
c) Adecuado porque cada objeto sabe y realiza su tarea.
d) Inadecuado porque el objeto :trayecto solo representa un trayecto.

a) La simplicidad es un criterio de evaluación de un diseño, pero este diseño NO es simple porque la distribución de responsabilidades no se corresponde con los roles de cada clase. Por ejemplo, el objeto :usuario no se debe ocupar de pedir el trayecto ni de ordenar el cálculo de la factura.

b) Cierto. El objeto :usuario tiene más responsabilidades de las que se corresponden con su rol de manejar aquello referente al usuario. Es un ejemplo de la escoba… expuesto en clase.

c) En cualquier diseño cada objeto realiza la tarea asignada, salvo alguna equivocación. Por tanto, el argumento dado no sirve para evaluar lo adecuado de un diseño.

d) Es adecuado que el objeto :trayecto se ocupe de un solo trayecto.



8. En el diseño de la figura anterior:

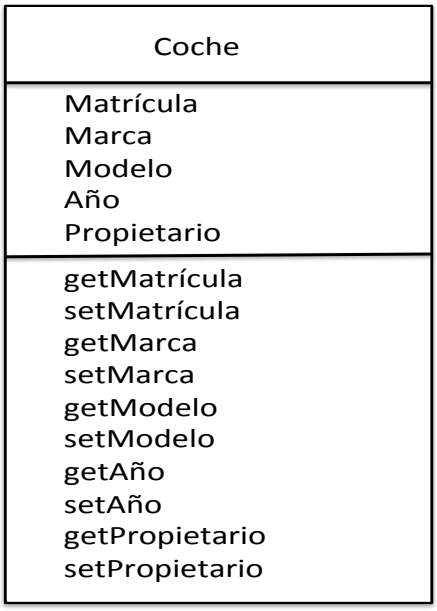
a) El método m1() debe cumplir el Principio de Sustitución.
b) El método n() no puede aprovechar la herencia.
c) El método m() de C1 cumple el Principio de Ocultación porque no se ve.
d) El método n() puede reutilizar el método m1().

a) Carece de sentido aplicar el Principio de Sustitución al método m1() porque no está en el contrato que ofrece la clase C.

b) Cierto. El método no puede aprovechar la herencia porque no tiene acceso a m1() y el método m() de C1, al que tiene acceso, es el mismo que el método homónimo de C.

c) El Principio de Ocultación no se relaciona con la invisibilidad física de nada.

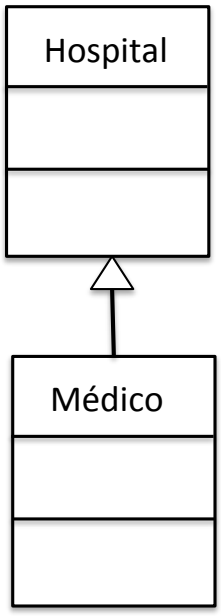
d) El término reutilizar es inapropiado en el argumento dado.



9. El diseño de la clase Coche es:

- **a) Adecuado porque intenta representar a un coche.**
- **b) Inadecuado porque se dedica a los datos del coche.**
- **c) Adecuado porque contiene la información de interés del coche.**
- **d) Inadecuado porque sus atributos son públicos.**

a) Los objetos software no representan a elementos del universo exterior al software.
b) Es inadecuado dedicar una clase solo a contener datos.
c) No es adecuado porque carece de los comportamientos que hacen de Coche un elemento activo
d) Carece de importancia que los atributos sean públicos si hay métodos para tomarlos y modificarlos



10. El diseño de la figura lateral es:

- **a) Correcto porque el médico está en el hospital.**
- **b) Correcto porque el médico pertenece al hospital.**
- **c) Incorrecto porque Médico debe tener menos métodos que Hospital.**
- **d) Incorrecto porque Médico no deriva de Hospital.**

La herencia software se interpreta solo en términos del verbo “ser”, ni “estar”, ni “pertenecer” atañen a la herencia software. Como médico no es un hospital, la respuesta acertada es la d).